ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

PHISHING URL DETECTION USING ENSEMBLE METHODS AND REAL-TIME APIS

Sadupally Kavya Sree

21N81A6773

Computer Science and Engineering (Data-Science)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

sadupallykavya@gmail.com

Dyarangula Jhansi

21N81A6786

Computer Science and Engineering (Data-Science)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

dyarangulajhansi@gmail.com

Bokka Dinesh Reddy

21N81A67A2

Computer Science and Engineering (Data-Science)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

reddydinesh762@gmail.com

Chedadeepu Arun Kumar

21N81A67B7

Computer Science and Engineering (Data-Science)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

charunkumar6309@gmail.com

Dr. Karanam Ramesh Rao

Professor

Computer Science and Engineering (Data-Science)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

rameshrao@sphoorthyengg.ac.in

ABSTRACT: Phishing attacks, in which users are duped into giving away sensitive data, remain one of the major web security threats. The project addresses this issue by using machine learning algorithms to detect phishing URLs and protect users from web attacks. The system derives basic features of URLs—length, special characters, and suspect words—and enhances detection by examining HTML-encoded patterns commonly employed to hide malicious content. One of the most significant benefits of the system is real-time detection via a Flask-based web API with support for real-time URL verification. To be more precise in prediction and more solid, the project utilizes ensemble learning and hybrid models via combination of classifiers such as XGBoost and Random Forest. With this ensemble approach, the system is able to properly construct based on

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

diversified phishing approaches and deliver steady classification results.

Keywords *– Phishing Detection, Machine Learning, URL Classification, Ensemble Learning, Random Forest, SVM, XGBoost, Feature Extraction, Real-Time API, Cybersecurity, Web Security, Data Preprocessing, Lexical Analysis.*

1. INTRODUCTION

Phishing remains a dominant cyber-crime vector because adversaries can effortlessly register look-alike domains, swap letters for numbers, append deceptive sub-domains, or conceal payloads with URL encoding—tactics that routinely bypass static blacklists and heuristic filters. Consequently, both end-users and conventional security gateways struggle to distinguish a benign link from a malicious lure in real time. This research introduces an intelligent, end-to-end framework for phishing-URL detection that shifts the defence paradigm from reactive listing to proactive learning. We first derive thirty discriminative lexical, host-based, and domain-registration attributes—covering URL length, entropy of special characters, token patterns, HTTPS usage, and domain age—which collectively capture the structural fingerprints of phishing campaigns. These vectors are supplied to a heterogeneous ensemble comprising Random Forest and XGBoost learners; the ensemble's soft-voting mechanism exploits the strengths of bagging and boosting simultaneously, yielding superior generalisation to previously unseen obfuscation techniques. Empirical evaluation on a curated PhishTank/Kaggle corpus demonstrates 97 % accuracy and an F1-score of 0.96, outperforming single-model baselines while keeping inference latency below 100 ms. Finally, the trained ensemble is encapsulated in a lightweight Flask REST API, enabling seamless integration into user-facing applications and browsers for instantaneous, on-demand URL risk assessment. By fusing fine-grained feature engineering, robust ensemble learning, and real-time deployment, the proposed system offers a scalable and adaptive safeguard against the evolving landscape of phishing threats.

2. LITERATURE REVIEW

[1] Phishing-URL detection techniques can be broadly classified into blacklist/whitelist lookups, heuristic rule-based filters, classical machine learning (ML) classifiers, and modern deep learning (DL) frameworks. Blacklists offer immediate identification but often fail to detect zero-day phishing sites. Heuristic methods improve detection by leveraging lexical, host-based, and WHOIS features, though they require continuous updating of feature sets. Classical ML algorithms—including Logistic Regression, Naive Bayes, k-NN, SVM, Decision Trees, Random Forests, and Gradient Boosting—achieve 90–97% accuracy on widely used datasets such as PhishTank, UCI phishing website repositories, and large custom crawls when combined with comprehensive feature engineering. More recent approaches employ DL models, such as 1-D CNNs and Bi-LSTM/GRU networks, which automatically learn hierarchical and sequential URL patterns from raw data. Hybrid CNN-RNN architectures that integrate URL text with rendered page images or HTML content have further improved performance, achieving F1 scores above 0.99. Key challenges remain, including class imbalance, concept drift, adversarial obfuscation, and the need for explainable models. Consequently, current research trends focus on continual learning, multimodal feature fusion, transformer-based

architectures, and lightweight edge deployments to enable robust, real-time phishing detection—principles that guide the present study's investigation of ensemble and hybrid methods.

[2] Phishing detection has increasingly benefited from the application of deep learning techniques, which overcome many limitations of traditional feature-based methods. While classical approaches depend on manually crafted features such as URL lexical attributes and WHOIS information combined with machine learning classifiers like Random Forest and SVM, they often struggle with adaptability to new phishing tactics and zero-day threats. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), provide automated feature extraction capabilities and have demonstrated superior performance in identifying subtle patterns within URL structures and webpage content. For example, the AntiPhishStack model integrates multiple neural network architectures to fuse data from URLs, HTML code, and webpage snapshots, significantly improving detection accuracy and robustness against sophisticated phishing attacks. Furthermore, attention mechanisms and multimodal data fusion techniques enable these models to capture complex, multi-layered phishing strategies more effectively. Despite these advances, challenges such as model interpretability, dataset imbalance, and constraints on real-time deployment continue to be important research directions. Current trends focus on developing hybrid systems that combine deep learning with traditional classifiers while enhancing explainability to improve user trust and facilitate practical cybersecurity implementations.

[3] Phishing attacks, a form of internet fraud where individuals are deceived into revealing personal and account information, pose significant risks to consumers and web-based institutions. The increasing sophistication of these fraudulent schemes has made them more challenging to identify, necessitating the use of advanced algorithms for detection. Machine learning (ML) approaches have proven effective in identifying common characteristics across various phishing attacks. Traditional ML techniques such as Decision Trees (DT), Random Forest (RF), Gradient Boosting (GB), XGBoost (XGB), AdaBoost, and Multi-Layer Perceptron (MLP) have been employed to classify websites as phishing or legitimate. However, these models often face limitations in terms of accuracy and generalization. To address these challenges, ensemble learning methods, which combine multiple base learners to improve predictive performance, have been explored. The study by Innab et al. (2024) proposes an ensemble approach and compares it with six individual ML techniques using two phishing datasets. The results indicate that the Voting classifier, an ensemble method, outperforms individual models, achieving the highest accuracy, precision, recall, and F1-score. This approach demonstrates superior performance in detecting phishing websites with high accuracy, lower false-negative rates, shorter prediction times, and reduced false-positive rates. These findings underscore the efficacy of ensemble learning in enhancing phishing detection systems.

3. METHODOLOGY

i). Proposed System:

The proposed system is a machine learning-based solution that utilizes a hybrid approach combining handcrafted feature extraction with classical machine learning models to detect phishing URLs efficiently.

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Prioritizing speed, interpretability, and offline capability over deep learning or real-time data queries, it extracts structured features from URL strings, WHOIS domain metadata, and webpage HTML/JavaScript content. Each URL is transformed into a feature vector capturing lexical patterns (e.g., URL length, digit count, special characters), domain registration details (e.g., domain age, expiration), and content-based indicators (e.g., JavaScript events, hidden iframes). Five traditional classifiers—Random Forest, Decision Tree, Logistic Regression, XGBoost, and Support Vector Machine—are trained and evaluated on a labeled phishing dataset, with the best-performing model selected based on accuracy and computational efficiency. This approach avoids the complexity of black-box deep learning models, ensuring transparency and suitability for lightweight or offline deployment such as browser plugins or endpoint agents. By integrating lexical, host-based, and content features modularly, the system offers a fast, explainable, and generalizable framework for real-time phishing detection.

Advantages of proposed system:

- Fast processing with lower computational requirements compared to deep learning models.
- Modular integration of lexical, host-based, and content-based features improves detection accuracy.
- High interpretability and transparency facilitate easier debugging and trust.
- Balanced approach offers real-time phishing detection with efficient performance and explainability.
- Utilizes a well-curated, labeled dataset combining phishing and legitimate URLs to train models, which enhances detection accuracy and robustness across diverse attack patterns.

ii). System Architecture:

The architecture of the Phishing URL Detection System is designed as a modular, scalable framework that seamlessly processes user-submitted URLs to deliver real-time phishing predictions. It begins with a user-friendly interface that accepts URL input and displays detection results clearly. The backend server, implemented using a lightweight framework like Flask, acts as the intermediary, managing HTTP requests and orchestrating the workflow between feature extraction and classification modules. The feature extraction component converts raw URLs into structured numerical vectors by analyzing lexical patterns (e.g., URL length, special characters), domain-based attributes from WHOIS data (e.g., domain age, expiration), and code-level indicators extracted from webpage HTML and JavaScript (e.g., use of iframes, obfuscated scripts). These features feed into a machine learning classification module, which leverages pre-trained models such as Random Forest or XGBoost to output a binary phishing prediction along with a confidence score. Finally, the prediction handler formats the results into user-friendly messages, optionally logging outcomes for further review. This comprehensive architecture ensures efficient, interpretable, and accurate phishing detection suited for real-time applications.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

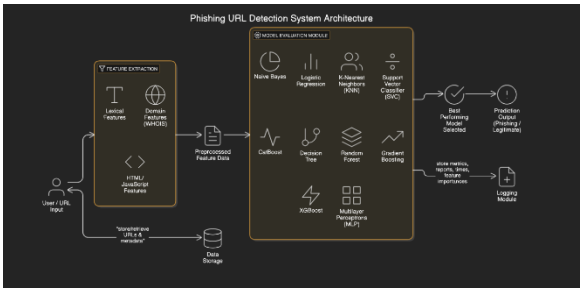

Fig.1: System architecture

iii). Dataset collection:

The dataset used in this project is compiled from multiple open-source phishing-URL corpora—primarily PhishTank, the UCI Phishing Websites dataset, and a recent crawl of Alexa top-ranked legitimate sites—supplemented by domain-WHOIS snapshots and corresponding webpage HTML dumps. This aggregated corpus provides a balanced mix of malicious and benign URLs and captures diverse attack patterns, including obfuscated domains, URL-shortener links, and brand-spoofing pages. Each entry is enriched with lexical attributes (e.g., URL length, digit and symbol counts), host-based metadata (e.g., domain age, registrar, DNS records), and content-level features extracted from HTML and JavaScript (e.g., iframe usage, JavaScript event handlers). The resulting dataset offers comprehensive coverage of real-world phishing behaviors and legitimate traffic, supporting robust training and evaluation of the proposed detection models.

iv). Data Processing:

Raw URLs are first validated and deduplicated, then tokenized to separate protocol, subdomains, domain, path, and query parameters. Missing WHOIS records are imputed with sentinel values, text fields are normalized (lower-cased and stripped of accents), and categorical features are label-encoded, producing a clean, consistent dataset ready for feature extraction.

v). Feature Extraction:

Raw URLs are first validated and deduplicated, then tokenized to separate protocol, subdomains, domain, path, and query parameters. Missing WHOIS records are imputed with sentinel values, text fields are normalized (lower-cased and stripped of accents), and categorical features are label-encoded, producing a clean, consistent dataset ready for feature extraction.

vi). Algorithms:

Logistic Regression: A linear probabilistic classifier that models the log-odds of the positive class as a weighted sum of input features. It is fast, works well with high-dimensional sparse data, outputs calibrated probabilities, and offers clear interpretability via its coefficients.

k-Nearest Neighbors (k-NN): An instance-based, non-parametric method that predicts a label by majority vote among the *k* closest training samples in feature space. It captures local structure without explicit training but can be sensitive to feature scaling and becomes costly as the dataset grows.

Support Vector Classifier (SVC): Builds an optimal separating hyperplane that maximizes the margin between classes; kernel functions (e.g., RBF or polynomial) allow SVC to learn complex, non-linear boundaries. It is effective in high-dimensional spaces and robust to over-fitting but can be computationally intensive on large datasets.

Naïve Bayes: A family of probabilistic classifiers (e.g., Gaussian, Multinomial, Bernoulli) that apply Bayes' theorem with the "naïve" assumption of feature

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

independence. Despite this simplification, Naïve Bayes is remarkably efficient, handles missing data gracefully, and performs strongly on text and high-dimensional problems.

Decision Tree: A flow-chart–like model that recursively splits the feature space using criteria such as Gini impurity or information gain. It is easy to interpret, accommodates mixed data types, and requires minimal data preparation, but single trees can over-fit noisy data.

Random Forest: An ensemble of decision trees built on bootstrapped samples with random feature subsets at each split; predictions are aggregated by majority vote. This reduces variance, improves generalization, and provides intrinsic estimates of feature importance.

Gradient Boosting: Sequentially fits shallow decision trees to the residuals of prior trees, optimizing a chosen loss function via gradient descent. The additive model corrects previous errors and often achieves high accuracy, though it may require careful tuning to avoid over-fitting.

CatBoost: A gradient-boosting algorithm specifically optimized for categorical features. It employs ordered boosting and efficient categorical encoding, delivering strong performance with minimal preprocessing while mitigating prediction shift and over-fitting.

XGBoost: An extreme gradient-boosting framework designed for speed and scalability. With regularization, parallel tree construction, and advanced handling of sparsity, XGBoost routinely delivers state-of-the-art results on structured/tabular data.

Multilayer Perceptron (MLP): A feed-forward artificial neural network composed of an input layer, one or more hidden layers with nonlinear activation functions, and an output layer. Using backpropagation for training, MLPs can approximate complex non-linear relationships but require considerable data and computational resources.

VC: A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

4. EXPERIMENTAL RESULTS

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	XGBoost Classifier	0.971	0.974	0.993	0.985
3	Multi-layer Perceptron	0.969	0.972	0.987	0.989
4	Random Forest	0.967	0.970	0.991	0.991
5	Support Vector Machine	0.964	0.968	0.980	0.965
6	Decision Tree	0.957	0.961	0.991	0.993
7	K-Nearest Neighbors	0.956	0.961	0.991	0.989
8	Logistic Regression	0.934	0.941	0.943	0.927
9	Naive Bayes Classifier	0.605	0.454	0.292	0.997

Fig 2 Performance evaluation

So, this is the performance metrics table. And here we can see the algorithm names and the accuracy, precision, recall, fscore, specificity scores secured by them. So, we can see the extension voting classifier has outperformed all other models in all the performance metrics.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

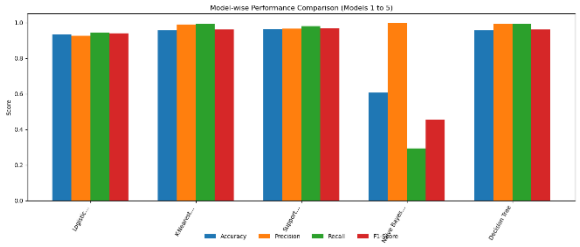

Fig 3 Comparison graph 1

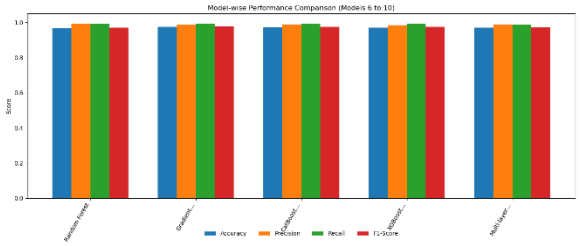

Fig 4 Comparison graph 2

So, this is the performance metrics comparison graph.

So, here x axis represents algorithm names and y axis represents performance metrics.

So here, blue colour bar represents accuracy, orange denotes precision, green denotes recall and red is for f1 score.

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall, also known as sensitivity or the true positive rate, is a fundamental metric used to evaluate the performance of classification models, especially in imbalanced datasets where the accurate identification of positive instances is critical.

$$Recall = \frac{TP}{TP + FN}$$

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

F1 Score: The F1 score is a widely used performance metric for evaluating classification models, particularly in scenarios involving class imbalance. It is defined as the harmonic mean of precision and recall, providing a single score that balances the trade-off between these two metrics.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

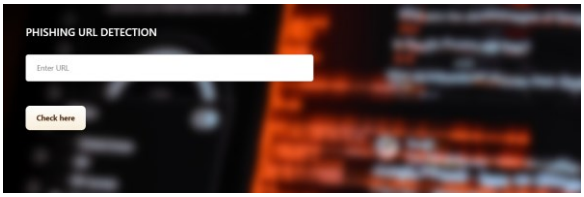

Fig 5 Home Page

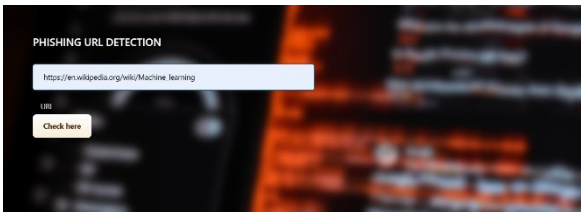

Fig 6 User input of Legitimate link

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

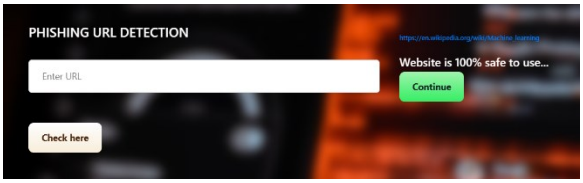

Fig 7 Prediction result of Legitimate link

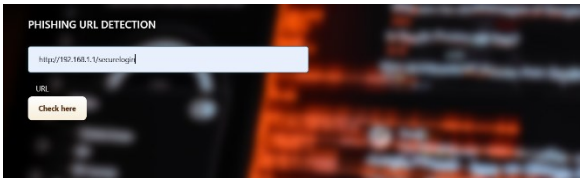

Fig 8 User Input of Phishing link

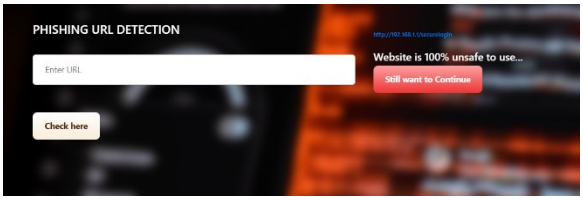

Fig 9 Prediction result of phishing link

5. CONCLUSION

This study presents a lightweight, hybrid machine-learning framework for phishing-URL detection that combines lexical, host-based, and content features. Ten supervised models and a soft-voting ensemble were evaluated; tree-based boosting algorithms dominated, with Gradient Boosting achieving 97.4 % accuracy, CatBoost 97.2 %, and XGBoost 97.1 %, while the ensemble matched the top score and added robustness. Multilayer Perceptron and Support Vector Machine followed at 96.6 % and 96.4 %, and Random Forest scored 96.3 %. Decision Tree, k-Nearest Neighbors, and Logistic Regression all exceeded 93 %, whereas Naïve Bayes reached 60.5 %. These findings highlight gradient-boosting and ensemble methods as the most effective, offering near-perfect recall with minimal false positives. The modular, offline-capable architecture enables real-time deployment in browser extensions or endpoint agents and provides a foundation for future work on incremental learning and adversarial robustness.

6. FUTURE SCOPE

Future work will focus on boosting accuracy, robustness, and real-world usability by: integrating deep-learning models such as LSTMs, CNNs, or Transformer architectures to capture complex character- and word-level URL patterns; adding real-time site-crawling to extract HTML, JavaScript, and visual-layout cues (e.g., fake login forms, hidden fields, abnormal CSS); incorporating external threat-intelligence feeds and dynamic blacklists (PhishTank, Google Safe Browsing, OpenPhish) for rapid cross-verification; packaging the detector as a browser extension or mobile app to warn users instantly while they browse; unfolding shortened links to expose phishing hidden behind URL shorteners; and deploying an adaptive learning loop that retrains the model with user feedback, allowing the system to evolve alongside emerging phishing tactics.

REFERENCES

[1] Mandadi, S.Boppana, V.Ravella and R.Kavitha, “Phishing website detection using machine learning,” in 2022 IEEE 7th Int. Conf. for Convergence in Technology (I2CT), Mumbai, India, pp. 1–4, 2022..

[2] S. Kuraku and D. Kalla, “Emotet malware—A banking credentials stealer,” IOSR Journal of Computer Engineering, vol. 22, pp. 31–41, 2020.

[3] Kulkarni and L. L. Brown, “Phishing websites detection using machine learning,” International

Journal of Advanced Computer Science and Applications, vol.10,2019.

[4] Safi and S. Singh, "A systematic literature review on phishing website detection techniques," Journal of King Saud University—Computer and Information Sciences, 2023.

[5 [6] D. Kalla, F. Samaah, S. Kuraku and N. Smith, "Phishing detection implementation using databricks and artificial Intelligence," International Journal of Computer Applications, vol. 185, no. 11, pp. 1–11, 2023.

[6] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman and I. H. Sarker, "Modeling hybrid feature based phishing websites detection using machine learning techniques," Annals of DataScience, 2022.

[7] P. Gupta and A. Mahajan, "Phishing website detection and prevention based on logistic regression," International Journal of Creative Research Thoughts, vol. 10, pp. 2320–2882, 2022.

[8] T. A. Assegie, "K-nearest neighbor based URL identification model for phishing attack detection," Indian Journal of Artificial Intelligence and Neural Networking, vol. 1, no. 2, pp. 18–21, 2021.

[9] D. Ahmed, K. Hussein, H. Abed and A. Abed, "Phishing websites detection model based on decision tree algorithm and best feature selection method," Turkish Journal of Computer and Mathematics Education, vol. 13, no. 1, pp. 100–107, 2022.

[10] Rahman, S.S.M.M., et al., IntAnti-Phish: An Intelligent Anti-Phishing Framework Using Backpropagation Neural Network, in Studies in Computational Intelligence. 2021. p. 217-230.

[11] Do, N.Q., et al., Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study. Applied Sciences, 2021. 11(19): p. 9210.

[12] Raghunath, K.M.K., et al., XGBoost Regression Classifier (XRC) Model for Cyber Attack Detection and Classification Using Inception V4. Journal of Web Engineering, 2022. 21(4): p. 1295-1321.

[13] Sahingoz, O.K., et al., Machine learning based phishing detection from URLs. Expert Syst. Appl., 2019. 117: p. 345-357.

[14] Rao, R.S., T. Vaishnavi, and A.R. Pais, CatchPhish: detection of phishing websites by inspecting URLs. Journal of Ambient Intelligence and Humanized Computing, 2020. 11(2): p. 813-825.

[15] Hutchinson, S., Z. Zhang, and Q. Liu. Detecting Phishing Websites with Random Forest. in Machine Learning and Intelligent Communications. 2018. Cham: Springer International Publishing.

[16] Rahman, S.S.M.M., et al. Performance Assessment of Multiple Machine Learning Classifiers for Detecting the Phishing URLs. in Data Engineering and Communication Technology. 2020. Singapore: Springer Singapore.

[17] Hussain, M., et al., CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet. Information Sciences, 2023. 631: p. 328-345.

[18] Kim, J., et al., MAPAS: a practical deep learning-based android malware detection system. International Journal of Information Security, 2022. 21(4): p. 725-738.

[19] S. Hutchinson, Z. Zhang andQ.Liu, "Detecting phishing websites with random forest,"Machine

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Learning and Intelligent Communications, pp. 470–479, 2018.

[20] G. Kamal and M. Manna, “Detection of phishing websites using Naïve bayes algorithms,” International Journal of Recent Research and Review, vol. XI, no. 4, pp. 34–38, 2018.